Reference Resolve

Your Rating: ☆☆☆☆☆ Results: ★★★★★ 10 rates

Current input

There are cases need to resolve:

1. General link field: `property = cafe-babe-cafee-babeee` or `property = /path/to/node`
    - ⚠ **Problem**: missing the workspace
    - ✓ **Ideas:**
        - *1st step:* This can change! We can suggest a new convention
            - `jcr:tours:cafe-babe-cafee-babeee`
            - `{resolverType}:{workspace}:{uuid}`
            - textImage00
                - text= prop
                - image =
                - source = jcr (String)
                - workspace = images (String)
                - uuid = cafe-babe (WeakReference)
        - *2nd step:* update linkField to use new convention (or customize reference type)
            - or ideally to reference a resolver in registry
2. Asset link field: `jcr:cafe-babe-cafee-babee` or `jcr:path/to/asset`
    - ⚠ **Problem**: Generate link with several renditions
3. Multi link field: `property = [cafe-babe-cafee-babee, aaa-bbbb-ccc-ddd]` or `property = [/path/to/node1, path/to/node2]`
    - ⚠ **Problem**: missing the workspace

Expect output

- Input: Node
    - linkedProperty = “jcr:cafe-babe-cafee-babeee"
- ReferenceResolverDefinition
- Output:
    - Node
        - linkedProperty (ReadOnlyNode)
            - Asset

        - name = cafe.jpg
        - path= /cafe.jpg
        - binary
        - mgnl:primaryType = mgnl:asset

Implementation

ReferenceResolver.java

```
package info.magnolia.reference;

import java.util.Optional;

@FunctionalInterface
interface ReferenceResolver<T, R> {
    Optional<R> resolve(T input);
}
```

FixedWorkspaceNodeReferenceResolver.java

```
/**
 * Sample implementation using a fixed target workspace (logical name).
 */
class FixedWorkspaceNodeReferenceResolver implements ReferenceResolver<String, Node> {
    private final String workspace;
    private final Provider<Context> contextProvider;

    public FixedWorkspaceNodeReferenceResolver(String workspace, Provider<Context> contextProvider) {
        this.workspace = workspace;
        this.contextProvider = contextProvider;
    }

    @Override
    public Optional<Node> resolve(String uuid) {
        try {
            Session session = contextProvider.get().getJCRSession(workspace);
            return Optional.of(session.getNodeByIdentifier(uuid));
        } catch (RepositoryException e) {
            e.printStackTrace();
            return Optional.empty();
        }
    }


        // Get/set workspace
}
```

Definition

Architecture discussion output

- Presented skeleton `ReferenceResolver` interface
    - is pretty much an empty shell;
    - can be implemented to resolve arbitrary items (*e.g.* `Node`, `Asset`), by arbitrary strategies (*e.g.* by uuid, by path)
- Agreed this is core functionality
- We first make it a standalone module
    - —for the lack of probation period to gain confidence with it, we want to have opportunity to make fast changes/releases if needed
    - repo in PLATFORM project
    - `info.magnolia.core:magnolia-reference:1.0-SNAPSHOT`
    - due to move back to the main reactor, eventually.

- Module provides a `Registry` of resolvers
    - Resolver selection yet to be specified
- New convention for a default (config-less) resolver to be determined, can be done later
    - would then be used for LinkFields by default
- Philip: which component is responsible for generating the reference format?
    - good remark, if not necessarily the role of the resolver, this seems to belong in this module.